【注意事項】

R20TS0750JJ0100
Rev.1.00
2021.10.01 号

M16C シリーズ、R8C ファミリ用 C/C++コンパイラパッケージ
M32C シリーズ用 C コンパイラパッケージ
R32C シリーズ用 C コンパイラパッケージ

概要

タイトルに記載している製品の使用上の注意事項を連絡します。

1. 単精度浮動小数点数の加減算をする場合の注意事項
2. 倍精度浮動小数点数の加減算をする場合の注意事項(1)
3. 倍精度浮動小数点数の加減算をする場合の注意事項(2)

1. 単精度浮動小数点数の加減算をする場合の注意事項

1.1 該当製品

- M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージ V.1.00 Release 1 ～ V.5.40 Release 00
- M32C シリーズ用 C コンパイラパッケージ V.1.00 Release 1 ～ V.5.42 Release 00
- R32C シリーズ用 C コンパイラパッケージ V.1.01 Release 00 ～ V.1.02 Release 01

1.2 内容

単精度浮動小数点数を加算、または減算した結果が特定の条件で±1.175494e-38 (単精度浮動小数点数で表現できる正規化数で、絶対値が最小の値)となる場合があります。（注 1）

注 1 : M32C シリーズ用 C コンパイラパッケージの場合には、特定の条件で±1.175494e-38 または ±0 となる場合があります。

1.3 発生条件

次の(1)から(3)のすべてを満たす場合に発生する可能性があります。

(1) 単精度浮動小数点数の加算、または減算をするコードを記述している

(2) (1)に対する出力コードがランタイムライブラリ関数__f4add または__f4sub の呼び出しである

(3) (1)の演算を加算で表現(a+b または b+a)した場合の符号部、指数部、仮数部と、該当製品、バージョンの関係、オプションが、以下の表を満たしている

●M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージ V.1.00 Release 1 ～ V.5.30 Release 02

条件 1 または条件 2 のいずれかを満たしている

条件	a と b の符号部	a の指数部	a の仮数部	b の指数部	b の仮数部
条件 1	異なる	0x19 から 0xFD	0x7FFFFF	a の指数部より 1 大きい値	0x000000
条件 2	異なる	0x18 から 0xFD	0x7FFFFE	a の指数部より 1 大きい値	0x000000

●M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージ V.5.40 Release 00

a と b の符号部	a の指数部	a の仮数部	b の指数部	b の仮数部
異なる	0x19 から 0xFD	0x7FFFFF	a の指数部より 1 大きい値	0x000000

●M32C シリーズ用 C コンパイラパッケージ V.1.00 Release 1 ～ V.5.42 Release 00

a と b の符号部	a の指数部	a の仮数部	b の指数部	b の仮数部
異なる	0x19 から 0xFD	0x7FFFFF	a の指数部より 1 大きい値	0x000000

-M82 オプション(※1)、-M90 オプション(※2) を使用した場合には、演算結果は±1.175494e-38 ではなく、±0 になる場合があります。

※1 -M82 オプションは、V.5.20 Release 1 以降で指定したときが該当します。

※2 -M90 オプションは、V.5.40 Release 00 以降で指定したときが該当します。

●R32C シリーズ用 C コンパイラパッケージ V.1.01 Release 00 ～ V.1.02 Release 01

a と b の符号部	a の指数部	a の仮数部	b の指数部	b の仮数部
異なる	0x19 から 0xFD	0x7FFFFF	a の指数部より 1 大きい値	0x000000

-fuse_FPU オプションを使用しない場合が該当します。

1.4 発生例

以下に発生例を示します。float の加減算の演算結果が、期待と異なり-1.175494e-38 になります。

```
/* tp.c */
float a = 127.999992;  /* 0x42FFFFFF (3)*/
float b = -128;        /* 0x43000000 (3)*/
float c;
 void f1(void) {
  c = a + b;  /* (1), (2) */
}
```

1.5 回避策

M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージをご使用の場合には、V.5.42 Release 00 以降をご使用ください。

演算結果の精度を落とすことを許容できる場合には、次の対応策により不具合の発生を防ぐことができます。

1.6 対応策

発生条件(3)を満たさない値に変更することで演算結果の違いを軽減することができます。以下に仮数部の下位 2 ビットをマスクして発生条件に該当しないようにする例を記します。

```
float a = 127.999992; /* 0x42FFFFFF */
float b = -128;       /* 0xC3000000 */
float c;

float maskTwoBit(float x) {
  union {
    float flt;
    unsigned long ul;
```

```
  } u;
  u.flt = x;
  if ((u.ul & 0x7FFFFEUL) == 0x7FFFFEUL) {
    u.ul &= 0xFFFFFFFCUL;
  }
  return u.flt;
}

void main(void) {
  a = maskTwoBit(a);   // 追加
  b = maskTwoBit(b);   // 追加
  c = a + b;
}
```

この例の場合の演算結果(c)は、-0.000000000000028421709430 となります。

1.7 恒久対策

改修の予定はありません。

2. 倍精度浮動小数点数の加減算をする場合の注意事項(1)

2.1 該当製品

- M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージ V.1.00 Release 00 ～ V.5.40 Release 00
- M32C シリーズ用 C コンパイラパッケージ V.1.00 Release 1 ～ V.5.40 Release 00

2.2 内容

倍精度浮動小数点数を加算、または減算した結果が、特定の条件を満たしたとき、演算結果の誤差が想定より大きくなる場合があります。特に、近い値を引き算して桁落ちが発生した場合に、最大で±1.00e-13 の誤差になります。

2.3 発生条件

次の(1)から(3)のすべてを満たす場合に発生する可能性があります。

(1) 倍精度浮動小数点数の加算、または減算をするコードを記述している

(2) (1)に対する出力コードがランタイムライブラリ関数__f8add または__f8sub の呼び出しである

(3) (3-1)から(3-3)のいずれかを満たす

(3-1) (1)の演算を加算で表現(a+b または b+a)した場合に、(3-1-1)と(3-1-2)のすべてを満たす

(3-1-1) a と b の符号および指数部が同じである

(3-1-2) a または b の仮数部において次に示すビットが 1 である

●M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージ の場合

1 ビット目(※)から 8 ビット目が 1 である

●M32C シリーズ用 C コンパイラパッケージ の場合

0 ビット目から 7 ビット目が 1 である

(※)“ビット目”の表記は、LSB を 0 ビット目として数えます。他の箇所の表記も同様です。

(3-2) (1)の演算を加算で表現(a+b または b+a)し、a の指数部は b の指数部より小さく、その差を d と表現した場合に、(3-2-1)と(3-2-2)のすべてを満たす

(3-2-1) d の値が、次のいずれかを満たしている

・1 から 7 の範囲内

・9 から 15 の範囲内

・17 から 23 の範囲内

・25 から 31 の範囲内

・33 から 39 の範囲内

・41 から 45 の範囲内

(3-2-2)　a または b の仮数部(けち表現を含む)において次に示すビットが 1 である

●M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージ の場合

d ビット目から(d+7)ビット目が 1 である

●M32C シリーズ用 C コンパイラパッケージ の場合

(d-1)ビット目から(d+7)ビット目が 1 である

(3-3)　(1)の演算を加算で表現(a+b または b+a)し、a の指数部は b の指数部より小さく、その差を d と表現した場合に、(3-3-1)と(3-3-2)のすべてを満たす

(3-3-1)　a と b の符号が同じである

(3-3-2)　a の仮数部にけち表現を含んだ値を a1m とし、b の仮数部にけち表現を含まない値を b0m と表現した場合に、((a1m >> d) + b0m)　または((a1m >> d) + b0m + 1)の計算結果が以下となる

●M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージ の場合

1 ビット目から 8 ビット目の 8 ビットと、52 ビット目が 1 である

●M32C シリーズ用 C コンパイラパッケージ の場合

0 ビット目から 8 ビット目の 9 ビットと、52 ビット目が 1 である

2.4 発生例

以下に発生例を示します。

```
void main(void) {
  /* 発生条件(3-1) */
  double a = 255.99999999999997; /* 0x406FFFFFFFFFFFF */
  double b = 128.0; /* 0x4060000000000000 */
  double c = a + b; /* (1)(2) */
  /* c は 384.000…(0x4078000000000000)が期待ですが、
   *      383.99999999998545(0x4077FFFFFFFFFF00) となります。 */

  /* 発生条件(3-2) */
  a = 127.999999999999992; /* 0x405FFFFFFFFFFFF */
  b = -128.0; /* 0xC060000000000000 */
  c = a + b; /* (1)(2) */
  /* c は -0.00000000000014210…(0xBD10000000000000)が期待ですが、
   *      -0.00000000007275957…(0xBDA0000000000000)となります    */

  /* 発生条件(3-3) */
  a = 127.99999999999955; /* 0x405FFFFFFFFFFE0 */
  b = 192.0000000000004; /* 0x406800000000000E */
  c = a + b; /* (1)(2) */
  /* c は 319.99999999999994(0x4073FFFFFFFFFFF)が期待ですが、
   *      319.99999999998545(0x4073FFFFFFFFFF00)となります    */
}
```

2.5 回避策

M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージをご使用の場合には、V5.42 Release 00 以降をご使用ください。

M32C シリーズ用 C コンパイラパッケージをご使用の場合には、V.5.41 Release 00 以降をご使用ください。

ただし、3. 倍精度浮動小数点数の加減算をする場合の注意事項(2)に該当する場合がありますので、あわせてご確認ください。

2.6 恒久対策

改修の予定はありません。

3. 倍精度浮動小数点数の加減算をする場合の注意事項(2)

3.1 該当製品

- M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージ V.5.42 Release 00 ～ V.6.00 Release 00
- M32C シリーズ用 C コンパイラパッケージ V.5.41 Release 00 ～ V.5.42 Release 00
- R32C シリーズ用 C コンパイラパッケージ V.1.01 Release 00 ～ V.1.02 Release 01

3.2 内容

倍精度浮動小数点数を加算、または減算した結果が、±2.2250738585072014e-308 (倍精度浮動小数点数で表現できる最小値)となる場合があります。

3.3 発生条件

次の(1)から(3)のすべてを満たす場合に発生する可能性があります。

(1) 倍精度浮動小数点数の加算、または減算をするコードを記述している

(2) (1)に対する出力コードがランタイムライブラリ関数__f8add または__f8sub の呼び出しである

(3) (1)の演算を加算で表現(a+b または b+a)した場合の符号部、指数部、仮数部と、該当製品、バージョンの関係が、以下の表を満たしている

●M16C シリーズ,R8C ファミリ用 C/C++コンパイラパッケージ V.5.42 Release 00 ～ V.6.00 Release 00

条件 1 または条件 2 のいずれかを満たす

条件	a と b の符号部	a の指数部	a の仮数部	b の指数部	b の仮数部
条件 1	異なる	0x36 から 0x7FD	0xFFFFFFFFFFFFF	a の指数部より 1 大きい値	0x0000000000000
条件 2	異なる	0x35 から 0x7FD	0xFFFFFFFFFFFFE	a の指数部より 1 大きい値	0x0000000000000

●M32C シリーズ用 C コンパイラパッケージ V.5.41 Release 00 ～ V.5.42 Release 00

a と b の符号部	a の指数部	a の仮数部	b の指数部	b の仮数部
異なる	0x36 から 0x7FD	0xFFFFFFFFFFFFF	a の指数部より 1 大きい値	0x0000000000000

●R32C シリーズ用 C コンパイラパッケージ V.1.01 Release 00 ～ V.1.02 Release 01

a と b の符号部	a の指数部	a の仮数部	b の指数部	b の仮数部
異なる	0x36 から 0x7FD	0xFFFFFFFFFFFFF	a の指数部より 1 大きい値	0x0000000000000

3.4 発生例

以下に発生例を示します。

```
/* tp.c */
double a = 127.99999999999999;  /* 0x405FFFFFFFFFFFF (3)*/
double b = -128;                /* 0xC060000000000000 (3)*/
double c;
void f1(void) {
  c = a + b;  /* (1), (2) */
}
```

この例の場合は、127.99999999999999(変数 a) + -128.0(変数 b) の演算結果(変数 c)が -0.00000000000001421085471520... となるべきですが、-2.2250738585072014e-308 になります。

3.5 対応策

発生条件(3)を満たさない値に変更することで演算結果の違いを軽減することができます。以下に仮数部の下位 2 ビットをマスクして発生条件に該当しないようにする例を記します。

```
double a = 127.99999999999999;
double b = -128;
double c;

double maskTwoBit(double x) {
  union {
    double dbl;
    unsigned long long ull;
  } u;
  u.dbl = x;
  if ((u.ull & 0xFFFFFFFFFFFFFEULL) == 0xFFFFFFFFFFFFFEULL) {
    u.ull &= 0xFFFFFFFFFFFFFFFCULL;
  }
  return u.dbl;
}

void main(void) {
  a = maskTwoBit(a);   // 追加
  b = maskTwoBit(b);   // 追加
  c = a + b;
}
```

この例の場合の演算結果(c)は、-0.00000000000000284217094304 となります。

3.6　恒久対策

改修の予定はありません。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Oct.01.21	-	新規発行

本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。

ニュース本文中の URL を予告なしに変更または中止することがありますので、あらかじめご承知ください。

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。

TS Colophon 4.2